Pose Estimation From a Single Image using Tensor Decomposition and an Algebra of Circulants

Randy C. Hoover, Karen S. Braman, and Ning Hao

Abstract— **Dimensionality reduction and object classification (recognition and pose estimation) serve as important tools in robotics, robotic vision, and industrial automation. The current paper presents a new approach to dimensionality reduction and object classification of three-dimensional rigid objects. The approach is based upon recent developments in tensor decompositions and a newly defined algebra of circulants. In particular, it is shown that under the right tensor multiplication operator, a third order tensor can be written as a product of third order tensors in which the left and right tensors are tensor-orthogonal and the inner-tensor is a diagonal tensor of singular-tuples. This new development allows for a proper tensor singular value decomposition (SVD) to be defined and has natural extension to tensor principal component analysis (PCA). Comparisons are made with traditional PCA and it is shown that the current approach is capable of recovering significantly more information from an image sequence using a much smaller subspace dimension. Further, it is shown that for most objects, accurate pose estimation can be performed from a single subspace dimension.**

I. Introduction

The classification of a three-dimensional (3-D) object from a two-dimensional (2-D) image has become an important issue in robotics, robotic vision, and industrial automation. Subspace methods have been successfully applied to many problems relating to classifying 3-D objects from 2-D images. Specific examples include face characterization and recognition [1]–[5], object recognition and pose estimation [6]–[15], as well as a host of applications which arise in industrial automation [16].

Subspace methods take advantage of the fact that a set of highly correlated images can be represented by a low-dimensional subspace [17]. Arguably the most popular linear subspace methods in use are principal component analysis (PCA), Fisher's linear discriminant analysis (FLDA), and locally preserving projections (LPP) [1]–[4], [7], [17]–[20]. Although PCA, FLDA, and LPP are unique subspace representations, as discussed in [15], all require the computation of the dominant principal components prior to classification.

Traditionally, the principal components of a large data-set have been computed using a linear algebraic framework. In particular, given a set of n images $\{I_1, I_2 \ldots, I_n\}$, each image is "row-scanned" into a column vector $\mathbf{x}$ of size $m = hv$. The collection of image vectors are then concatenated to form an "image data matrix" $X = [\mathbf{x}_1, \mathbf{x}_2 \cdots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$ where in general $m >> n$. The principal components are found by utilizing the singular value decomposition (SVD), i.e., $SVD(X) = U\Sigma V^T$ where U and V are orthogonal and Σ is a diagonal matrix of singular values. The principal components of X are found to be the columns of U with the most dominant components corresponding to the largest singular values in Σ.

Although matrix PCA has shown success in the past, the natural representation of each image as a matrix is lost by vectorizing these images for the construction of X. A more natural representation for the collection of images is to leave each image in matrix form, and stack each matrix into a tensor structure (also referred to as an n-way or n-mode array). As illustrated in [21], the tensor structure immediately lends itself to multi-linear algebraic operations on high-order tensors. Although there is no *unique* SVD-like decomposition for high-order tensors, several tensor-SVD solutions have been proposed that attempt to mimic similar properties of the matrix SVD [21]–[27]. While the N-mode SVD outlined in [21] has shown promising results for tensor PCA and ICA, the decomposition is constructed from products of matrices with a core-tensor as opposed to a product of tensors as one might expect.

In the current paper, we describe a new approach for computing the principal components of an n-way array using a high-order SVD (referred to as the t-SVD throughout the remainder of this paper). The approach follows recent developments based upon Fourier theory and an algebra of circulants as outlined in [28]–[32]. It is shown that under the right tensor multiplication operator, a third order tensor can be written as a product of third order tensors in which the left- and right-singular tensors are tensor-orthogonal and the inner-tensor is a front-face diagonal (denoted as f-diagonal) tensor of singular-tuples. In the context of robotic vision, we illustrate that for most objects, accurate pose estimation can be performed using a single subspace dimension. The subspace in this case is the left-singular matrix associated with the largest singular-tuple of the resulting t-SVD.

The remainder of this paper is organized as follows: In Section II we discuss the relevant tensor algebra, the newly defined tensor multiplication operator, and the t-SVD. In Section III we present an analysis of how the proposed tensor PCA is different from traditional PCA, and discuss some of the advantages and capabilities of the tensor PCA. Section IV provides an example of the current approach to pose estimation of objects using a single subspace dimension.

Randy C. Hoover is with Department of Electrical and Computer Engineering, South Dakota School of Mines and Technology, Rapid City, SD 57701, USA randy.hoover@sdsmt.edu

Karen Braman is with the Department of Mathematics, South Dakota School of Mines and Technology, Rapid City, SD 57701, USA karen.braman@sdsmt.edu

Ning Hao is a Ph.D. student in the Department of Mathematics, Tufts University, Medford, MA 02155, USA ning.hao@tufts.edu

This section also presents an error analysis associated with the current approach compared to that of traditional PCA. Finally, Section V presents some concluding remarks and provides some insight into future research directions.

II. Mathematical Foundations of Tensors

A. Mathematical Preliminaries

The term *tensor*, as used in the context of this paper, refers to a multi-dimensional array of numbers, sometimes called an *n-way* or *n-mode* array. If, for example, $\mathcal{A} \in \mathbb{R}^{\ell\times m\times n}$ then we say $\mathcal{A}$ is a third-order tensor where *order* is the number of ways or modes of the tensor. Thus, matrices and vectors are second-order and first-order tensors, respectively. Fundamental to the results presented in this paper is a recently defined multiplication operation on third-order tensors which itself produces a third-order tensor [28], [29].

Further, it has been shown by Braman [30] that under this multiplication operation, $\mathbb{R}^{\ell\times m\times n}$ is a free module over a commutative ring with unity where the "scalars" are $\mathbb{R}^{1\times 1\times n}$ tuples. In addition, it has been shown in [30] and [31] that all linear transformations on the space $\mathbb{R}^{\ell\times m\times n}$ can be represented by multiplication by a third-order tensor. Thus, even though $\mathbb{R}^{\ell\times m\times n}$ is not strictly a vector space, many of the familiar tools of matrix linear algebra can be applied in this new context, including the basic building blocks of principal component analysis. For a more in depth discussion on this topic, the reader is referred to [30].

First, we review the basic definitions from [29] and [28] and introduce some basic notation. It will be convenient to break a tensor $\mathcal{A}$ in $\mathbb{R}^{\ell\times m\times n}$ up into various slices and tubal elements, and to have an indexing on those. The i^{th} lateral slice will be denoted $\mathcal{A}_i$ whereas the j^{th} frontal slice will be denoted $\mathcal{A}^{(j)}$. In terms of MATLAB indexing notation, this means $\mathcal{A}_i \equiv \mathcal{A}(:,i,:)$ while $\mathcal{A}^{(j)} \equiv \mathcal{A}(:,:,j)$.

We use the notation $\mathbf{a}_{ik}$ to denote the i,k^{th} tube in $\mathcal{A}$; that is $\mathbf{a}_{ik} = \mathcal{A}(i,k,:)$. The j^{th} entry in that tube is $\mathbf{a}_{ik}^{(j)}$. Indeed, these tubes have special meaning for us in the present work, as they will play a role similar to scalars in $\mathbb{R}$. Thus, we make the following definition:

Definition 1: An element $\mathbf{c} \in \mathbb{R}^{1\times 1\times n}$ is called a **tubal-scalar** of length n.

As mentioned previously, the set of tubal-scalars with length n endowed with element-wise addition and tensor multiplication (defined by the t-product in Def. 2) forms a commutative ring [30]. For ease of notation, we will use $\mathbf{0}$ to denote the additive identity, i.e., the tubal-scalar with all zero elements. Let $\mathbf{e}_1$ denote the tubal-scalar with all zero elements except a 1 in the first position. Then it is easy to see that $\mathbf{e}_1$ is the multiplicative identity in this ring and it will play an important role in the remaining tensor definitions.

In order to discuss multiplication between two tensors we must first introduce the concept of converting $\mathcal{A} \in \mathbb{R}^{\ell\times m\times n}$ into a block circulant matrix.

If $\mathcal{A} \in \mathbb{R}^{\ell\times m\times n}$ with $\ell\times m$ frontal slices then

$$\texttt{circ}(\mathcal{A}) = \begin{bmatrix} A^{(1)} & A^{(n)} & A^{(n-1)} & \dots & A^{(2)} \\ A^{(2)} & A^{(1)} & A^{(n)} & \dots & A^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A^{(n)} & A^{(n-1)} & \ddots & A^{(2)} & A^{(1)} \end{bmatrix},$$

is a block circulant matrix of size $\ell n \times mn$.

We anchor the MatVec command to the frontal slices of the tensor. $\texttt{MatVec}(\mathcal{A})$ takes an $\ell\times m\times n$ tensor and returns a block $\ell n\times m$ matrix

$$\texttt{MatVec}(\mathcal{A}) = \begin{bmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \\ A^{(n)} \end{bmatrix}.$$

The operation that takes $\texttt{MatVec}(\mathcal{A})$ back to tensor form is the fold command:

$$\texttt{fold}(\texttt{MatVec}(\mathcal{A})) = \mathcal{A}.$$

With these two operations in hand, we are in a position to introduce the t-product between two, third-order tensors that was developed in [28], [29]:

Definition 2: Let $\mathcal{A} \in \mathbb{R}^{\ell\times p\times n}$ and $\mathcal{B} \in \mathbb{R}^{p\times m\times n}$ be two third order tensors. Then the **t-product** $\mathcal{A} * \mathcal{B} \in \mathbb{R}^{\ell\times m\times n}$ is defined as

$$\mathcal{A} * \mathcal{B} = \texttt{fold}(\texttt{circ}(\mathcal{A}) \cdot \texttt{MatVec}(\mathcal{B})\ ).$$

Note that in general, the t-product of two tensors will not commute. There is one special exception in which the t-product always commutes: the case when $\ell = p = m = 1$, that is, when the tensors are tubal-scalars.

Definition 3: The **identity tensor** $\mathcal{I} \in \mathbb{R}^{m\times m\times n}$ is the tensor whose frontal slice is the $m\times m$ identity matrix, and whose other frontal slices are all zeros.

Definition 4: If $\mathcal{A}$ is $\ell\times m\times n$, then the **tensor transpose** $\mathcal{A}^T$ is the $m\times \ell\times n$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through n.

In order to build the concept of orthogonality in the space $\mathbb{R}^{\ell\times m\times n}$, a bilinear form is defined on $\mathbb{R}^{m\times 1\times n}$ as follows [29], [31], [32]: suppose that $\mathcal{X},\mathcal{Y} \in \mathbb{R}^{m\times 1\times n}$, then

$$\langle \mathcal{X},\mathcal{Y}\rangle = \mathcal{X}^T * \mathcal{Y}. \qquad (1)$$

It is easy to show that this operation satisfies the properties of conjugate symmetry and linearity. Using this bilinear form, orthogonality is a straight forward analog of the vector definition.

Definition 5: Let $\mathcal{X}$ and $\mathcal{Y}$ be elements of $\mathbb{R}^{m\times 1\times n}$. Then $\mathcal{X}$ and $\mathcal{Y}$ are called **orthogonal** if

$$\langle \mathcal{X},\mathcal{Y}\rangle = \mathbf{0}, \qquad (2)$$

where $\mathbf{0} \in \mathbb{R}^{1\times 1\times n}$ is the additive identity.
We can also define orthogonality for tensors.

Definition 6: A tensor $\mathcal{X} \in \mathbb{R}^{\ell \times m \times n}$ is called orthogonal if

$$\mathcal{X} * \mathcal{X}^T = \mathcal{X}^T * \mathcal{X} = \mathcal{I},$$

where $\mathcal{I}$ is the identity tensor.

Definition 7: The tensor norm used throughout this paper is the Frobenious norm which for the tensor $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$ is given by

$$\|\mathcal{A}\|_F = \sqrt{\sum_{i=1}^{\ell} \sum_{j=1}^{m} \sum_{k=1}^{n} a_{ijk}^2}. \tag{3}$$

where a_{ijk} is the i, j, k^{th} element of $\mathcal{A}$.

B. Computation of the t-SVD

The final tool necessary for tensor PCA is a tensor singular value decomposition. In [28], [29] the authors show that, for $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$, there exists orthogonal tensors $\mathcal{U}$ and $\mathcal{V}$, and an f-diagonal tensor $\mathcal{S}$ such that

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T = \sum_{i=1}^{\min(\ell,m)} \mathcal{U}_i * \mathbf{s}_i * \mathcal{V}_i^T, \tag{4}$$

where the $\mathcal{U} \in \mathbb{R}^{\ell \times \ell \times n}$ is the tensor of left-singular matrices, $\mathcal{S} \in \mathbb{R}^{\ell \times m \times n}$ is an f-diagonal tensor of singular tuples, $\mathcal{V} \in \mathbb{R}^{m \times m \times n}$ is the tensor of right-singular matrices, and $\mathbf{s}_i = \mathcal{S}(i, i, :)$ are the singular tuples. Throughout this paper, we refer to this decomposition as the **t-*SVD***. A graphical illustration of the t-*SVD* is shown in Fig. 1.

Computation of the t-*SVD* comes from the constructive proof outlined in [28], [29], which will be restated here for completeness. It is well known in matrix theory that a circulant matrix can be diagonalized via left and right multiplication by a discrete Fourier transform (DFT) matrix. Similarly, a block circulant matrix can be block diagonalized via left and right multiplication by a block diagonal DFT matrix. For example, consider the tensor $\mathcal{A} \in \mathbb{R}^{\ell \times m \times n}$, then

$$(F_n \otimes I_\ell)\texttt{circ}(\mathcal{A})(F_n^* \otimes I_m) = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_n \end{bmatrix},$$

where each of the D_i are $\ell \times m$, $I_{(\cdot)}$ is an identity matrix of dimension $(\cdot)$, F_n is the $n \times n$ DFT matrix, F_n^* is its conjugate transpose, and $\otimes$ is the Kronecker product. To construct the t-*SVD* defined in (4), the matrix SVD is performed on each of the D_i, i.e., $D_i = U_i S_i V_i^T$ resulting in the decomposition shown in (5). Applying $(F_n^* \otimes I_n)$ to the left and $(F_n \otimes I_n)$ to the right of each of the block diagonal matrices on the right hand side of (5) results in each being block circulant, i.e., if we define $\hat{U}$ as the block diagonal matrix with U_i as its diagonal blocks, then

$$(F_n^* \otimes I_\ell)\hat{U}(F_n \otimes I_\ell) = \begin{bmatrix} U_1 & U_n & \cdots & U_{n-1} \\ U_2 & U_1 & \cdots & U_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ U_n & U_{n-1} & \dots & U_1 \end{bmatrix}.$$

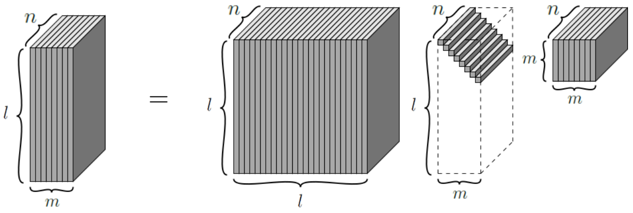

Fig. 1. Graphical illustration of the t-SVD of an $l \times m \times n$ tensor.

Taking the first block column of each block circulant matrix and applying the `fold` operator results in the decomposition $\mathcal{U} * \mathcal{S} * \mathcal{V}^T$. Note that for simplicity, as well as computational efficiency, this entire process can by performed using the fast Fourier transform in place of the DFT matrix as illustrated in [28], [29].

III. An Analysis of Tensor Decomposition Applied to Image Sequences

A. Image Acquisition

This section provides an overview of the how image sequences are acquired and preprocessed prior to analysis. The current work considers gray-scale images described by an $h \times v$ array of square pixels with intensity values normalized between 0 and 1. Thus an image is represented by a matrix[1] $I \in [0, 1]^{h \times v}$. In particular, we consider image sequences where an object is placed at the center of an imaging sphere, as seen in Fig. 2, and a sample image is acquired at each of the black dots on the surface of the sphere. Setting β to a constant results in a one-dimensionally correlated sequence of images. The collection of images $\{I_1, I_2 \dots, I_n\}$ are then laterally concatenated to form the image data-tensor $\mathcal{X} \in \mathbb{R}^{h \times n \times v}$ as shown in Fig. 3.

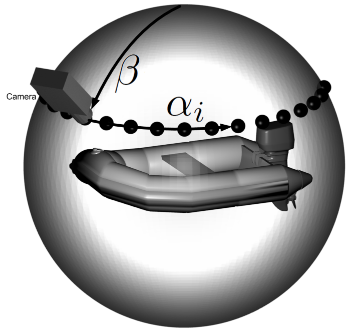

Fig. 2. This figure depicts sampling an object along a line of constant co-latitude, which results in a sequence of images correlated in a single dimension. A sample image is captured at each of the black dots that reside on the imaging sphere.

To illustrate the concepts described in Section II as they apply to image analysis, $n = 128$ images of objects being

[1] Note that the image matrix I should not be confused with the standard identity matrix.

$$\begin{bmatrix} D_1 & & \\ & \ddots & \\ & & D_n \end{bmatrix} = \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_n \end{bmatrix} \begin{bmatrix} S_1 & & \\ & \ddots & \\ & & S_n \end{bmatrix} \begin{bmatrix} V_1^T & & \\ & \ddots & \\ & & V_n^T \end{bmatrix} \tag{5}$$

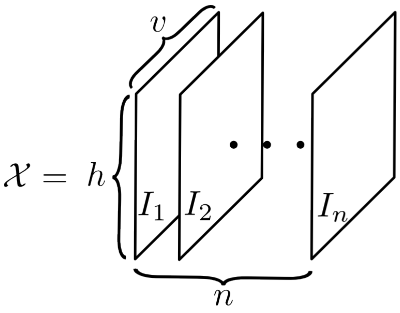

Fig. 3. A graphical representation of how the image data-tensor $\mathcal{X}$ is constructed from the set of n images $\{I_1, I_2 \ldots, I_n\}$.

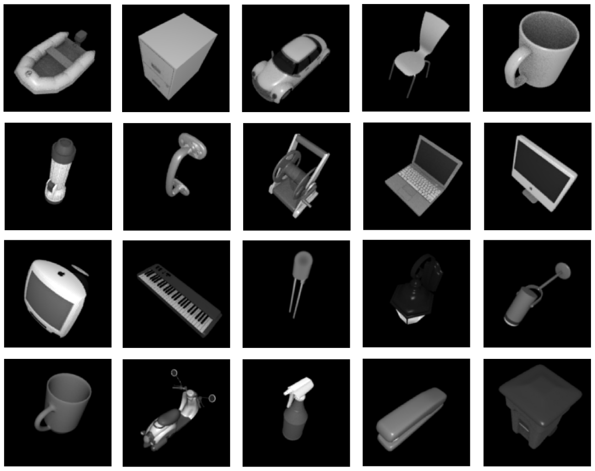

Fig. 4. Images of Ray-traced CAD models courtesy of [33]. Images of each object are captured as discussed in Section II at a resolution of 128 × 128. The objects are ordered from left to right, then top to bottom.

rotated about a single degree of freedom were captured in the manner described above with $\beta = 60°$ measured down from the upper pole. Each image was of size $h = v = 128$. The set of images were then concatenated to construct the image data-tensor $\mathcal{X} \in \mathbb{R}^{128\times128\times128}$. The images of each object were generated by ray-tracing CAD models provided by [33]. An example image of each object is depicted in Fig. 4. The t-SVD was then applied to $\mathcal{X}$ to generate the left-singular matrices $\mathcal{U}$. For comparison, we also computed the matrix PCA of the image data set as described in Section I, taking the principal components as the left-singular vectors of U. Note that in the literature, the PCA decomposition is commonly referred to as eigenspace decomposition and the resulting left-singular matrices and left-singular vectors are referred to as eigenimages. We too will adopt this terminology with the eigenimages being the left-singular vectors of U and the t-eigenimages being the left-singular matrices of $\mathcal{U}$.

B. Information Recovery

The naive approach to object recognition and pose estimation using appearance-based methods is to compare a query image to each image in the training set to estimate the closest match. Unfortunately, for large data sets, this approach is computationally infeasible. Therefore, we require that the dimensionality of the image set be reduced for efficient comparison. It has been shown in [1]–[3], [7]–[9], [11], [14], [15], [18] that because the image data set is highly correlated, most of the information in the entire set of images can be recovered by projecting the image data onto a much smaller subspace. Therefore, we denote by $\mathcal{U}_k$ as the tensor containing the first k left-singular matrices of $\mathcal{U}$ and U_k as the matrix containing the first k left-singular vectors of U. In the current analysis, $\mathcal{U}_k \in \mathbb{R}^{128\times k\times128}$ whereas $U_k \in \mathbb{R}^{128^2\times k}$. Dimensionality reduction is performed using matrix PCA by projecting the image data onto the first k left singular vectors associated with the largest k singular values, i.e., $M_k = X^T U_k \in \mathbb{R}^{n\times k}$. Similarly, the tensor counterpart is computed using the t-product as $\mathcal{M}_k = \mathcal{X}^T * \mathcal{U}_k \in \mathbb{R}^{128\times k\times128}$. To determine what subspace dimension k is sufficient to capture enough information about our image data set, we rely on the energy recovery ratio defined for matrix PCA in [9] as

$$\rho(X, U_k) = \frac{\sum_{i=1}^{k}(\sigma_i^2)}{\|X\|_F^2}, \tag{6}$$

where σ_i is the i^{th} singular value of X. Note that the energy recovery ratio achieves a maximum value of one as $k \to n$. A similar measure can be defined for tensor PCA, however, recall that the "singular values" for tensors are actually singular-tuples $\mathbf{s}_i$. Therefore, the one subtle difference between the matrix energy recovery and the tensor counterpart is the Frobenius norm of the singular-tuple as well as the tensor $\mathcal{X}$, i.e.,

$$\text{t-}\rho(\mathcal{X}, \mathcal{U}_k) = \frac{\sum_{i=1}^{k}\|\mathbf{s}_i\|_F^2}{\|\mathcal{X}\|_F^2}. \tag{7}$$

Similar to its matrix counterpart, the tensor energy recovery ratio t-$\rho(\cdot)$ achieves a maximum value of one as $k \to n$.

One of the most notable advantages tensor PCA has over its matrix counterpart is the capability of recovering a significant amount of energy at a much lower subspace dimension. To illustrate this concept, both the matrix and tensor energy recovery ratio were computed for each of the 20 objects in Fig. 4 as a function of the subspace dimension k. The results of this analysis are depicted in Fig. 5. As is apparent from the figure, on average, the tensor PCA recovers over 95% of the information in $\mathcal{X}$ with only a $k = 5$-dimensional subspace whereas the matrix PCA requires on average a $k = 15$-dimensional subspace for the

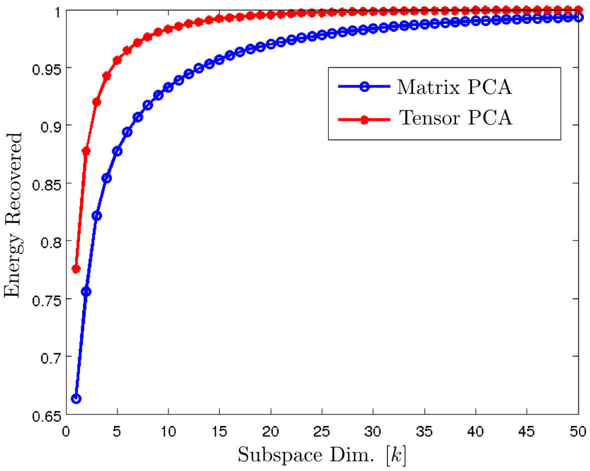

Fig. 5. Energy recovery ratio ρ and t-ρ averaged across all 20 objects in Fig. 4. As is evident from the figure, the energy recovered by the tensor PCA is significantly larger than its matrix counterpart for a similar subspace dimension.

same information recovery. While this could be a result of the singular tuples containing more information than a singular value, this verification is left for future work.

Another interesting observation is the resulting eigenimages computed by each of the decompositions. The first nine eigenimages and t-eigenimages of object 5 in Fig. 4 are depicted in Fig. 6. The top row in Fig. 6 shows the t-eigenimages and the bottom row shows the eigenimages. In [9], it was shown that for one-dimensionally correlated image sequences, the right-singular vectors are well approximated by Fourier harmonics. This result is visually evident in the bottom row of Fig. 6 as the eigenimages "appear" to contain harmonics of increasing frequency from left to right. A similar harmonic pattern can be seen in the t-eigenimages in Fig. 6, however, we notice that the tensor PCA actually distributes these harmonics throughout the spatial domain horizontally. While this observation is evident graphically, future work will focus on a quantification of these ideas.

IV. An Application to Pose Estimation

A. Introduction

This section presents a practical application of the developments in Section II as applied to robotic systems. Traditional PCA has been successfully applied to estimating the pose (spatial orientation) of 3-D objects in many different problem domains [6]–[15]. For its simplicity and ease of visualization, we first discuss the matrix version of pose estimation using PCA. This discussion will then be extended to illustrate the the tensor version of PCA in the following section.

B. Pose Estimation via PCA

1) Matrix PCA: Estimating the pose of 3-D objects using PCA is done in two separate phases, namely, the training phase and the estimation phase. The training phase is completed off-line and consists of constructing the image data matrix X, and computing the principal eigenimages U_k, where U_k consists of the first k columns of U. Once the principal eigenimages of an image data matrix have been computed, the dimensionality of the image data is reduced using the projection techniques discussed in Section III-B, i.e., $M_k = X^T U_k$. This projection generates a set of n points in the eigenspace, each point having dimension k. Assuming the object's pose is the only variable changing to generate X, this set of points is a discrete approximation to the underlying one-dimensional manifold embedded in k-dimensional space.

The on-line estimation phase then consists of projecting a query image into the eigenspace. Because a projection of an image of the object will likely be close to the eigenspace manifold computed from training images of the same object, the pose of the object can be estimated by determining the closest point on the manifold, typically using a Euclidean distance measure.

2) Tensor PCA: The extensions of pose estimation using tensor PCA are analogous to their matrix counterparts. The training phase consists of constructing the image data tensor $\mathcal{X}$ as described in Section III and computing the principal t-eigenimages $\mathcal{U}_k$ using equation (4). Similarly, dimensionality reduction is performed using the techniques described in Section III-B to construct $\mathcal{M}_k = \mathcal{X}^T * \mathcal{U}_k$. Note however that unlike the matrix counterpart, $\mathcal{M}_k$ is not necessarily a discrete approximation to a one-dimensional manifold. This is due to the fact that for tensors, the projection results in a set of k tuples rather than k scalars.

The on-line estimation phase using tensor PCA needs some special attention as well. The projection of a query image I_q into the t-eigenspace is done in a similar manner. However, I_q must first be "transformed" into a $h \times 1 \times v$ tensor prior to using the t-product for the projection. Another subtle difference between tensor PCA and its matrix counterpart is the comparison of this projection with the projection of each of the training images to obtain an estimate of the pose. In matrix PCA, both projections produce a point in the eigenspace, therefore, a simple Euclidean norm can be used as a distance metric. In tensor PCA however, the comparison of each projection results in a set of k n-dimensional tuples that must be ordered. It has been shown in [32] that this ordering can be performed by taking the 1-norm of each of the tuples. The estimate of the pose is then determined by finding the index that corresponds to the minimum of this ordering.

C. Experimental Results

1) Pose Estimation Analysis: To illustrate the concepts of pose estimation using matrix and tensor PCA, similar to Section III, $n = 128$ images of objects being rotated about a single degree of freedom were captured in the manner described in Section III-A with $\beta = 60°$ measured down from the upper pole. Each image was of size $h = v = 128$. For matrix PCA, each image is then "row-scanned" to construct the image vector $\mathbf{x} \in \mathbb{R}^{128^2 \times 1}$. The set of image vectors were then concatenated to construct the image data matrix X. For tensor PCA, the set of images were concatenated to construct the image data-tensor $\mathcal{X} \in \mathbb{R}^{128 \times 128 \times 128}$, where each image forms a lateral slice of the tensor as illustrated in Fig. 3. To

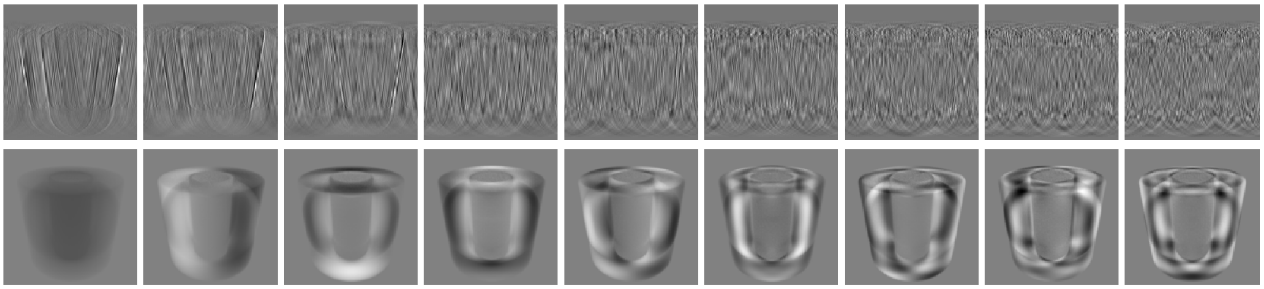

Fig. 6. The first nine t-eigenimages [top] and eigenimages [bottom] of object 5 in Fig. 4 as computed by the t-SVD and the matrix SVD respectively.

evaluate the accuracy of estimation, an additional 64 query images of each object were captured at random but known poses in a similar fashion to that illustrated in Fig. 2.

For both techniques, the left-singular vectors (U_k) or matrices ($\mathcal{U}_k$) were computed from X and $\mathcal{X}$ respectively. The pose of each object was then estimated for each of the 64 query images using a subspace dimension $k = 3$ and $k = 1$. Table I tabulates the results of this experiment for each of the 20 objects. Columns 2-5 illustrate the energy recovered using traditional PCA as well as t-PCA for both subspace dimensions. As can be seen from the table, for the same subspace dimension, t-PCA is capable of recovering more information for each of the 20 objects. Columns 6-9 show the absolute error in pose estimation averaged across all 64 query images for both subspace dimensions. Column 6 shows that using traditional PCA, a subspace dimension of $k = 1$ is not sufficient to estimate the pose of any of the objects tested. However, as shown in column 8, with the exception of objects 2 and 13 t-PCA is capable of accurately estimating the pose of each object with a single subspace dimension. Also note that increasing the subspace dimension from $k = 1$ to $k = 3$ is enough for t-PCA to accurately estimate the pose of each object while PCA still has issues for 7 of the 20 objects. Finally, the last column in Table I shows the on-line time required for accurate pose estimation to be performed using the t-PCA with a subspace dimension $k = 1$, averaged across all 64 query images. While not as fast as typical matrix PCA, this still meets most real-time robotics requirements.

2) Analysis of Objects 2 and 13 for $k = 1$: This subsection briefly addresses the miss-classified poses for objects 2 and 13 in column 8 of Table I. Notice that for a subspace dimension $k = 1$ using t-PCA, the average absolute pose estimation error for object 2 across all 64 query images was 6.92°. Upon further analysis of this object, it was determined that 3 of the 64 images were miss-classified, thereby increasing the average estimation error. The three miss-classified images are a result of an ill-posed classification problem due to a possible modulo π rotation. In fact, the classification error was determined to be off by nearly 180° for each of the three query images. A similar analysis held true for the miss-classified images for object 13 as well. Therefore, we conclude that for most objects, accurate pose estimation can be obtained by a single subspace dimension using the current approach.

V. Conclusions and Future Directions

This paper presented a new approach to dimensionality reduction and object classification using tensor decompositions and an algebra of circulants. In particular, it was shown that using a newly defined tensor multiplication operator, a third order tensor can be written as a product of third order tensors in which the left and right tensors are tensor-orthogonal and the inner-tensor is f-diagonal tensor of singular-tuples. This decomposition is analogous to the matrix SVD and thereby lends itself to a new tensor PCA. An analysis was presented in the context of robotic vision and it was shown that the new tensor decomposition is very promising for dimensionality reduction and object classification. In fact, it was shown that for the objects tested, a single subspace dimension was sufficient to accurately estimate the pose of the object for nearly all query images. This new tensor SVD opens up many exciting branches for future research directions in several different areas of robotics, computer vision, and image processing. Most pressing will be extensions to independent component analysis (ICA), LDA, and locally preserving projections (LPP).

References

[1] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Trans. PAMI*, vol. 12, no. 1, pp. 103–108, Jan. 1990.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, Mar. 1991.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. PAMI*, vol. 19, no. 7, pp. 711–720, July 1997.

[4] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *IEEE Conf. Comp. Vis. and Patt. Rec.*, Seattle, WA, June 1994, pp. 84–91.

[5] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. PAMI*, vol. 24, no. 1, pp. 34–58, Jan. 2002.

[6] H. Murase and S. K. Nayar, "Illumination planning for object recognition using parametric eigenspaces," *IEEE Trans. PAMI*, vol. 16, no. 12, pp. 1219–1227, Dec. 1994.

[7] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance," *Int. J. Comp. Vis.*, vol. 14, no. 1, pp. 5–24, Jan. 1995.

[8] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace methods for robot vision," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 750–758, Oct. 1996.

[9] C. Y. Chang, A. A. Maciejewski, and V. Balakrishnan, "Fast eigenspace decomposition of correlated images," *IEEE Trans. Image Proc.*, vol. 9, no. 11, pp. 1937–1949, Nov. 2000.

[10] R. C. Hoover, A. A. Maciejewski, and R. G. Roberts, "Pose detection of 3-D objects using S^2-correlated images and discrete spherical harmonic transforms," in *IEEE Int. Conf. Robot. Automat.*, Pasadena, CA, May. 2008, pp. 993–998.

TABLE I

THE ENERGY RECOVERED USING PCA AND T-PCA FOR SUBSPACE DIMENSIONS $k = 1$ AND $k = 3$, THE ABSOLUTE ERROR IN POSE ESTIMATION AVERAGED ACROSS 64 QUERY IMAGES USING BOTH TECHNIQUES FOR BOTH SUBSPACE DIMENSIONS, AS WELL AS THE AVERAGE ON-LINE TIMING REQUIREMENTS FOR ESTIMATING THE POSE USING THE T-PCA AND $k = 1$.

Obj.	ρ		t-ρ		Ave. Error [deg.] (PCA)		Ave. Error [deg.] (t-PCA)		t-PCA timing [sec.]
	$k = 1$	$k = 3$	$k = 1$	$k = 3$	$k = 1$	$k = 3$	$k = 1$	$k = 3$	at $k = 1$
1	0.73	0.84	0.77	0.92	79.26	4.05	1.53	1.51	0.261
2	0.91	0.96	0.95	0.98	101.83	21.47	6.92	1.53	0.265
3	0.64	0.88	0.74	0.93	85.79	20.99	1.51	1.51	0.266
4	0.53	0.79	0.70	0.92	81.98	1.43	1.47	1.51	0.265
5	0.82	0.91	0.91	0.97	38.08	1.51	1.51	1.51	0.266
6	0.64	0.85	0.73	0.92	101.86	1.50	1.55	1.53	0.265
7	0.49	0.76	0.67	0.90	81.37	1.55	1.51	1.51	0.265
8	0.67	0.76	0.75	0.87	61.14	1.90	1.53	1.53	0.266
9	0.73	0.84	0.78	0.92	56.59	1.54	1.53	1.51	0.266
10	0.64	0.76	0.76	0.91	71.66	1.57	1.53	1.55	0.266
11	0.63	0.73	0.72	0.87	58.06	7.19	1.55	1.51	0.266
12	0.40	0.61	0.53	0.77	42.55	1.53	1.53	1.53	0.266
13	0.75	0.96	0.93	0.99	94.16	96.20	4.29	1.55	0.270
14	0.72	0.85	0.85	0.95	47.23	1.48	1.48	1.51	0.267
15	0.27	0.55	0.52	0.82	36.13	1.49	1.53	1.51	0.267
16	0.81	0.93	0.93	0.98	46.70	1.47	1.51	1.51	0.267
17	0.50	0.70	0.69	0.88	73.71	6.79	1.51	1.51	0.272
18	0.87	0.93	0.89	0.97	108.78	1.97	1.57	1.53	0.272
19	0.60	0.84	0.75	0.93	64.25	1.51	1.51	1.51	0.267
20	0.93	0.97	0.95	0.98	107.63	23.90	1.51	1.51	0.271

[11] K. Saitwal, A. A. Maciejewski, and R. G. Roberts, "Computationally efficient eigenspace decomposition of correlated images characterized by three parameters," *Patt. Anal. and App.*, vol. 12, no. 4, pp. 391 – 406, Oct. 2009.

[12] R. C. Hoover, A. A. Maciejewski, and R. G. Roberts, "Pose detection of 3-D objects using images sampled on $SO(3)$, spherical harmonics, and Wigner-D matrices," in *IEEE Conf. on Automat. Sci. and Engr.*, Washington DC, Aug 2008, pp. 47–52.

[13] R. C. Hoover, "Pose estimation of spherically correlated images using eigenspace decomposition in conjunction with spectral theory," Ph.D. dissertation, Colorado State University, 2009.

[14] R. C. Hoover, A. A. Maciejewski, and R. G. Roberts, "Eigendecomposition of images correlated on S^1, S^2, and $SO(3)$ using spectral theory," *IEEE Trans. Image Proc.*, vol. 18, no. 11, pp. 2562–2571, Nov. 2009.

[15] R. C. Hoover, A. A. Maciejewski, and R. G. Roberts, "Fast eigenspace decomposition of images of objects with variation in illumination and pose," *IEEE Tran. Sys. Man, Cyber. B: Cybernetics*, vol. PP, no. 99, pp. 1–12, Aug. 2010.

[16] E. N. Malamasa, E. G. Petrakisa, M. Zervakisa, L. Petit, and J.-D. Legat, "A survey on industrial vision systems, applications and tools," *Image and Vision Comp.*, vol. 21, no. 2, pp. 171–188, 2003.

[17] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. London, U.K.: Academic, 1990.

[18] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer.*, vol. 4, no. 3, pp. 519–524, Mar. 1987.

[19] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. PAMI*, vol. 7, no. 3, pp. 328–340, Mar. 2005.

[20] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal Laplacianfaces for face recognition," *IEEE Trans. Image Proc.*, vol. 15, no. 11, pp. 3608–3614, Nov. 2006.

[21] M. Alex, O. Vasilescu, and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *European Conf. on Comp. Vis.*, Copenhagen, Denmark, May 2002, pp. 447 – 460.

[22] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sept. 1966.

[23] R. A. Harshman, "Foundations of the PARFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis," University of California Los Angelas, Tech. Rep. 10,085, December 1970.

[24] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, March 2000.

[25] O. Vasilescu and D. Terzopoulos, "Multilinear projection for appearance-based recognition in the tensor framework," in *Int. Conf. on Comp. Vis.*, 2007, pp. 1–8.

[26] O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *Int. Conf. on Comp. Vis. and Patt. Rec.*, 2003, pp. 93–99.

[27] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognition*, vol. 44, no. 7, pp. 1540–1551, July 2011.

[28] M. E. Kilmer, C. D. Martin, and L. Perrone, "A third-order generalization of the matrix SVD as a product of third-order tensors," Tufts University, Department of Computer Science, Tech. Rep. TR-2008-4, October 2008.

[29] M. E. Kilmer and C. D. Moravitz Martin, "Factorization strategies for third-order tensors," *Linear Algebra and Its Applications*, no. Special Issue in Honer of G.W.Stewart's 75^{th} birthday, 2009.

[30] K. Braman, "Third-order tensors as linear operators on a space of matrices," *Linear Algebra and its Applications*, vol. 433, no. 7, pp. 1241 – 1253, 2010.

[31] M. Kilmer, K. Braman, and N. Hao, "Third order tensors as operators on matrices: A theoretical and computational framework," Tufts University, Department of Computer Science, Tech. Rep. TR-2011-01, January 2011.

[32] D. F. Gleich, C. Greif, and J. M. Varah, "The power and arnoldi methods in an algebra of circulants," *arXiv*, vol. 1101.2173v1, 2011.

[33] (2007) Kator Legaz: 3-D model database for Blender. [Online]. Available: http://www.katorlegaz.com/